MSFT Docket No. 155593.1
Attorney Docket No. MCS-057-00

SYSTEM AND METHOD FOR HIGH-DENSITY INTERACTIVE VOTING USING A COMPUTER NETWORK

by

Jong S. Huang
Ian Marriott
Curtis N. von Veh

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to interactive voting using a computer network and more particularly to a system and method for facilitating high-density interactive voting (such as during a live event) using a computer network and providing voting results in real-time.

2. Related Art

Real-time polls using interactive voting over a computer network are popular for determining audience response to a particular survey question. Usually the real-time poll is conducted in connection with a live event such as a television or radio broadcast. The real-time poll and associated survey question typically coincides with a featured story being presented during the live event. Real-time polls using interactive voting are especially popular with live events such as television news magazine programs because such polls make television programs more interactive and give viewers an opportunity to voice their opinions in real time. In general, interactive voting involves presenting survey question and a number of responses to a voter (usually a member of the live event audience), determining a voter's response, and tabulating and presenting voting results. Real-time polls using interactive voting are becoming more pervasive due to the convergence of the television and the computer. Although the technology is available so that the voter can directly interact and vote using the television, the most common way for the voter to interactively respond to a survey question is by the voter using a computer connected to a computer network (such as the Internet).

By way of example, a sponsor of the live event (such as a television news magazine program) may conduct a real-time poll by posing a survey question to its audience, presenting a number of responses and asking the audience to vote on the survey question by visiting the sponsor's web site. The voter uses her computer to visit the sponsor's web site and selects a response to the survey question by clicking on one of the options. When the voter has finished voting she submits the response to the web site and the vote is transmitted from the user's computer to the web server at the sponsor's site. The web server receives the voter's vote, adds the vote to other votes received and tabulates the voting results. The voting results are posted as part of the story for the voter to observe how other people voted in comparison to her own vote. Because the television or radio broadcast is usually live (hence the phrase "live event") it is desirable for the sponsor to be able to conduct the entire real-time poll within the time constraints of the live event (typically one-hour in duration).

One interactive voting system and method is described in U.S. Patent No. _____ (Serial No. 09/064,567) by West et al. entitled "System and Method for Interactive Online Voting". This patent describes a database system for tracking votes cast by voters in response to an interactive poll and returning voting results. The voters cast votes by over a network (such as the Internet) and the voting results are returned to the voters. In general, a voter visits a web site containing the survey question and submits their vote to the web site. When the vote is submitted, the voting system opens a web page (such as an Active Server Page) that receives the vote and then transmits the vote to a database. The web page opens a connection to the database each time a voter submits a vote and then closes the connection. One problem with this type of interactive voting system, however, is that it was not designed to handle large volumes of votes within a short period of time (i.e., high-density voting). For example, this system typically can handle only about six votes per second, mainly because the database connections must be opened and closed each time a vote is submitted. This repeated opening and closing of database connections imposes a heavy burden on the web site server when there are a large number of voters in a short amount of time. In addition, the server is often overwhelmed which cause a malfunction or even shutdown of the computer system (also known as "crashing").

Accordingly, there exists a need for an interactive voting system and method using a computer network that is capable of handling high-density voting. What is also needed is a high-density interactive voting system and method that allows live television programs to be more interactive by allowing viewers of these television programs a chance to voice their

opinions to a survey question and obtain the results in real time. What is further needed is a high-density interactive voting system and method that addresses the problem of repeated opening and closing of database connections whenever a large volume of votes is received in a short period of time.

SUMMARY OF THE INVENTION

To overcome the limitations in the prior art as described above and other limitations that will become apparent upon reading and understanding the present specification, the present invention includes system and method for facilitating high-density interactive voting over a network and providing voting results in real-time. The present invention facilitates real-time polling by providing a utility that aids and simplifies the creation of survey and demographic questions. Generally, these survey questions are provided during a live event (such as a television or radio broadcast) and responses and results are required within a short amount of time. This creates a high voting density on the network that can incapacitate an interactive voting network. The present invention uses novel features to maximize throughput over the voting network and allow high-density voting. In addition, the present invention is able to tabulate and return the voting results in real-time, thereby enabling the results of the voting to be broadcast during the live event.

In general, the high-density interactive voting system of the present invention includes five major components that reside on the voting network: (1) a Live Event Wizard; (2) a Live Event Staging Component; (3) a Live Event Database; (4) a Live Event Object (LEO); and (5) a Live Event Display Engine. The Live Event Wizard enables a user to quickly and easily create polling questions (such as survey and demographic questions) for distribution and presentation to a voter. The Live Event Staging Component ensures the validity of the survey and demographic questions and sends a copy of a live event definition to the Live Event Database. The Live Event Database (which may be, for example, based on Microsoft Structured Query Language (SQL)) contains all the database tables and stored procedures to maintain information and integrity related to the votes. Moreover, the Live Event Database is responsible for tabulating current voting results at specified (or pre-determined) time intervals, and performing statistical calculations of voting results such as, by way of example, performing cross tabulation of questions versus demographics. The LEO is a server side object (and preferably may be used on Active Server Pages) that pools or combines connections to the Live Event Database, caches the votes received from voters and performs batch vote processing before sending the batch results to the Live Event

Database. The Live Event Display Engine receives the live event definition from the LEO and renders the definition and results on a display device for voters to view.

The method of the present invention includes receiving votes from voters, tabulating the received votes at a specified time interval to generate intermediate voting results, caching the intermediate results in memory, and using the intermediate results to compute a final voting result. Receiving the votes from voters is accomplished using the LEO that resides in memory (such as random access memory (RAM)). In addition, the LEO maintains persistent connections with the Live Event Database. In other words, database connections are maintained and separate connections are not opened and closed each time a voter transmits a vote. Preferably, there are three persistent connections maintained: (1) the current voting results; (2) the raw vote cast by each voter; and (3) the live event definition.

Other aspects and advantages of the present invention as well as a more complete understanding thereof will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention. Moreover, it is intended that the scope of the invention be limited by the claims and not by the preceding summary or the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention can be further understood by reference to the following description and attached drawings that illustrate the preferred embodiments. Other features and advantages will be apparent from the following detailed description of the invention, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the present invention.

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a block diagram illustrating a computing apparatus that preferably may be used to carry out the present invention.

FIG. 2 is an overall block diagram of computer network using the computing apparatus of FIG. 1 and incorporating the high-density interactive voting system of the present invention.

FIG. 3 is a general block diagram illustrating the components of the high-density interactive voting system of FIG. 2.

FIG. 4 is a general flow diagram of the operation of the high-density interactive voting system shown in FIGS. 2 and 3.

FIG. 5 is a detailed flow diagram illustrating the operation of the Live Event Wizard and the Live Event Staging Component of the Live Event Authoring System shown in FIG. 3.

DETAILED DESCRIPTION OF THE INVENTION

In the following description of the invention, reference is made to the accompanying drawings, which form a part thereof, and in which is shown by way of illustration a specific example whereby the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

I. Introduction

The present invention includes a high-density interactive voting system and method for facilitating high-density interactive voting (such as during a live event) using a computer network and providing voting results in real-time. In general, the present invention receives a vote over a computer network from a voter in response to a survey question, adds that vote to the other votes received and tabulates the votes to provide real-time voting results. In addition, the present invention is capable of handling high-density voting that is required in order to allows a large number of voters to vote in a short period of time and to obtain final voting results in real-time. Previous interactive voting system and techniques make direct database connections from a Web page (using, for example, an Active Server Page (ASP) page). Moreover, these techniques create a new and separate connection for each voter. Thus, each time a voter casts a vote a connection with the database is established and opened, the vote is written to the database and the connection is closed. During periods of high-density voting this continual opening and closing of database connections can severely strain and even shutdown the network.

The present invention alleviates this problem by providing a Live Event Object (LEO) that resides and remains in memory (such as random access memory (RAM)). The LEO is a self-contained entity that contains voting data and procedures and instructions for manipulating the voting data. Moreover, the LEO maintains persistent connections with a Live Event Database so that these connections do not have to continually be opened and closed. These persistent connections greatly reduce the burden on the server and increase system performance. In other words, maintaining these persistent connections significantly reduces the amount of processor time, bandwidth and memory needed to communicate with the Live Event Database. Moreover, the LEO caches in memory votes received from a large

number of voters and, at a predefined interval, tallies and sends the intermediate voting results to the Live Event Database where the final results are generated from these intermediate voting results. Vote caching as performed by the present invention allows these intermediate voting results to be tabulated continuously to generate final voting results much faster than can be obtained by tabulating each vote individually. Thus, another advantage of the present invention is that, unlike previous interactive voting techniques that tabulate results after all the votes have been received, the present invention computes intermediate voting results at specified intervals to enable rapid and real-time tabulation of final voting results.

II. Exemplary Operating Environment

FIG. 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the high-density interactive voting system and method of the present invention may be implemented. Although not required, the present invention will be described in the general context of computer-executable instructions (such as program modules) being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with a variety of computer system configurations, including personal computers, server computers, hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located on both local and remote computer storage media including memory storage devices.

With reference to FIG. 1, an exemplary system for implementing the present invention includes a general-purpose computing device in the form of a conventional personal computer 100, including a processing unit 102, a system memory 104, and a system bus 106 that couples various system components including the system memory 104 to the processing unit 102. The system bus 106 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 110 and random access memory (RAM) 112. A basic input/output system (BIOS)

114, containing the basic routines that help to transfer information between elements within the personal computer 100, such as during start-up, is stored in ROM 110. The personal computer 100 further includes a hard disk drive 116 for reading from and writing to a hard disk (not shown), a magnetic disk drive 118 for reading from or writing to a removable magnetic disk 120, and an optical disk drive 122 for reading from or writing to a removable optical disk 124 (such as a CD-ROM or other optical media). The hard disk drive 116, magnetic disk drive 128 and optical disk drive 122 are connected to the system bus 106 by a hard disk drive interface 126, a magnetic disk drive interface 128 and an optical disk drive interface 130, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 100.

Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 120 and a removable optical disk 124, it should be appreciated by those skilled in the art that other types of computer readable media that can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read-only memories (ROMs), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 120, optical disk 124, ROM 110 or RAM 112, including an operating system 132, one or more application programs 134, other program modules 136 and program data 138. A user (not shown) may enter commands and information into the personal computer 100 through input devices such as a keyboard 140 and a pointing device 142. In addition, other input devices (not shown) may be connected to the personal computer 100 including, for example, a microphone, joystick, game pad, satellite dish, scanner, and the like. These other input devices are often connected to the processing unit 102 through a serial port interface 144 that is coupled to the system bus 106, but may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor 146 or other type of display device is also connected to the system bus 106 via an interface, such as a video adapter 148. In addition to the monitor 146, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The personal computer 100 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 150. The remote computer 150 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the

elements described above relative to the personal computer 100, although only a memory storage device 152 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 154 and a wide area network (WAN) 156. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 100 is connected to the local network 154 through a network interface or adapter 158. When used in a WAN networking environment, the personal computer 100 typically includes a modem 160 or other means for establishing communications over the wide area network 156, such as the Internet. The modem 160, which may be internal or external, is connected to the system bus 106 via the serial port interface 144. In a networked environment, program modules depicted relative to the personal computer 100, or portions thereof, may be stored in the remote memory storage device 152. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

III. General Overview

The invention is embodied in a method and a system for facilitating high-density interactive voting over a network and obtaining voting results in real time. FIG. 2 is an overall block diagram of computer network using the computing apparatus of FIG. 1 and incorporating the high-density interactive voting system of the present invention. It should be noted that the high-density interactive voting system 200 illustrated is only one of several ways in which the present invention may be implemented.

In general, the high-density interactive voting system 200 includes a Live Event Authoring System 205 for facilitating the creation of survey and demographic questions 210 for a real-time poll. Typically, the Live Event Authoring System 205 will be located at the broadcast studios 215 of the sponsor of the real-time poll (such as a television network). A user (not shown) interacts with the Live Event Authoring System 205 through a Live Event Authoring System User Interface 220. In FIG. 2, the Live Event Authoring System 205 and the Live Event Authoring System User Interface 220 are shown as implemented on a computer (such as the computing apparatus of FIG. 1), although other implementations will be evident to those skilled in the art. Through the Live Event Authoring System User Interface 220, the user writes and edits survey questions related to the real-time poll and possible responses to the questions. In addition, the user can create demographic

questions to determine how demographics influence results of the real-time poll. Many types of real-time polls may be conducted by generating a variety of questions and responses using the Live Event Authoring System 205.

The survey and demographic questions 210 are sent to television transmission facilities 225 for broadcast during a live event. In general, the real-poll is conducted during the live event and relates to the topic of discussion featured on the live event. For example, a television news magazine program may be featuring a story on the Electoral College, and one of the survey questions may be whether the Electoral College should be abolished. A viewer that is viewing the live event broadcast receives the broadcast by a television 230. At an appropriate time during the live event broadcast, the survey questions are displayed to a viewer through a voting interface 235 (such as a television screen) and the viewer is asked to participate in the real-time poll by logging on to a web site (usually the web site of the live event sponsor). For example, during a live television new magazine program the on-air personality will instruct viewers to log onto a web site and participate in the online real-time poll on a topic that is being presented by the program. The present invention tabulates and displays the results of this real-time poll instantly.

If a viewer decides to participate in the real-time poll, the viewer uses a client computer 240 to access the designated web site. The client computer 240 and a Live Event Vote Server 245 communicate over a network 250 and form a computer network. The survey and demographic questions 210 are been sent to the Live Event Vote Server 245 and are displayed on the web site for the voter to view. On the client computer 240, the voter is able to view the survey and demographic questions 210 and cast votes using a High-Density Interactive Voting System Voter Interface 255 of the present invention. Once the voter cast a vote, a voter input 260 is transmitted over the network 250 to the Live Event Vote Server 245. As discussed in detail below, a High-Density Interactive Voting Module 265 resides on the Live Event Vote Server 245 and efficiently receives and tabulates the results of the voting in real-time. These vote results 270 are sent to the client computer 240 over the network 250 for display to the voter through the High-Density Interactive Voting System Voter Interface 255. In addition, the vote results 270 are sent to the broadcast studios 215 for transmission 225 to the voter's television 230 and the vote results 270 are displayed by the voting interface 235. Preferably, the vote results 270 are tabulated and transmitted prior to the conclusion of the live event broadcast so that viewers can see the results of the real-time poll.

IV. Components and Operation of the Invention

FIG. 3 is a general block diagram illustrating the components of the high-density interactive voting system of FIG. 2. In general, the voting system includes a Live Event Authoring System 205, a Live Event Vote Server 245 and a client computer 240. The Live Event Authoring System 205 enables a live event author 300 to define an event and create polling questions associated with the event. The Live Event Authoring System 205 includes a Live Event Wizard 305 and a Live Event Staging Component 310. The Live Event Wizard 305 is a tool for the Live Event Author 300 to use that enables the rapid creation of survey and demographic questions associated with a real-time poll. The Live Event Staging Component 310 checks the validity of the survey and demographic questions to ensure that no errors or incongruities are present and also copies the event definition to a Live Event Database 315.

The Live Event Vote Server 245 is part of a computer network and is in communication with a plurality of computers, as represented by the client computer 240. In addition, it should be noted that the Live Event Vote Server 245 may include a plurality of servers such that the entire high-density interactive voting system of the present invention is scalable by using either one or more servers. By way of example, the voting density of a single server is approximately 70 votes/second while the voting density across a plurality of servers will increase in direct proportion to the number servers deployed. Therefore, if a high voting density is anticipated additional servers may be added to the Live Event Vote Server 245 to enable high-density voting.

The Live Event Vote Server 245 is in communication with the Live Event Authoring System 205, the client computer 240 and a television broadcast 320. A voting environment 325, which is the environment where a voter interacts with the high-density interactive voting system of the present invention, includes television viewers 330 and computer network users (or voters) 335. The television broadcast 320 includes an event and question display 340 for displaying the survey questions to the television viewers 330 and a voting results display 345 for displaying voting results to the viewers 330. The voter (who is also a television viewer 330) views the survey questions on a television (not shown), decides to vote and then accesses the Live Event Vote Server 245 through the appropriate web site. Other voters also access the Server 245 and each of these computer network users 335 access the web site and connect to the Server 245 using their individual client computer 240.

The Live Event Vote Server 245 includes a Live Event Database 315, a Live Event Display Engine and a Live Event Object 355. The Live Event Database 315, which is preferably based on Microsoft Structured Query Language (SQL), contains all the database tables and stored procedures to maintain voting-related information and integrity. In addition, the Live Event Database 315 receives and stores polling questions and event definitions created by the Live Event Authoring System 205 and stores voting results. The Live Event Display Engine 350 includes a survey question display module 360 for obtaining event definition and polling questions from the Live Event Object 355 and the Live Event Database 315 and rendering them on the client computer 240. Moreover, the Live Event Display Engine 350 includes a Voter Input Module 365 for receiving raw votes from each voter, and a Voting Results Display Module 370 for obtaining voting results from the Live Event Database 315 and rendering them on the client computer 240. In addition, the voting results are sent to the television broadcast 320 so that the television viewers 330 can view the voting results.

At the heart of the Live Event Vote Server 245 is the Live Event Object (LEO) 355. The LEO 355 is a server-side object that pools database connections, caches votes received and performs batch processing prior to sending the batch results to the Live Event Database 315. The LEO 355 resides in memory 375 and includes a voting cache 380 used for caching votes. When an event is defined and the first votes are received concerning the event, the LEO 355 makes persistent connections 385 with the Live Event Database 315. These connections are persistent because once they are established they remain open and do not close once an individual vote has been sent to the Live Event Database 315. The LEO 355 receives a vote from the Voter Input Module 365 and caches the vote along with other received votes in the Voting Cache 380. At a predefined interval the votes within the Voting Cache 380 sent to the Live Event Database 315 and are tabulated using a Live Event Vote Processor 390 to generate intermediate voting results. Preferably, a plurality of intermediate voting results are generated and stored in the Live Event Database 315. In order to determine final voting results, the Live Event Vote Processor 390 is used to tally the intermediate voting results to arrive at a final voting result.

In a preferred embodiment, the LEO 355 maintains three persistent connections with the Live Event Database 315 including an Event Definition 391, a Raw Vote 392 and Current Voting Results 393. When an event definition is sent to the Live Event Vote Server 245 and posted on the web site, the Event Definition 391 persistent connection is used by LEO 355 to load the event. LEO 355 is responsible for delivering survey content (questions

and allowed responses) and tabulations (current voting results and percentages) as needed to, for example, an Active Server Page (ASP). The LEO 355 accomplishes this by establishing and maintaining a Raw Vote 392 persistent connection with the Live Event Database 315 for sending accumulated vote tabulations from the Voting Cache 320 to the Live Event Database 315. Moreover, the LEO 355 establishes and maintains a Current Voting Results 393 persistent connection to the Live Event Database 315 to query the Live Event Database 315 for the updated voting results and deliver these results to the Live Event Display Engine 350 and the television broadcast 320. If the persistent connections 385 are broken for some reason then LEO 355 will attempt to re-establish the connections. The persistent connections 385 are closed only when the LEO 355 is shut down.

The LEO 355 manages a list of current event definitions on which a voter can vote. These event definitions are constructed by running a set of instructions (or Stored Procedure) the first time that they are requested by a client. Subsequently, the event definitions are cached in memory until they expire. By default, all event definitions are removed from the event list after a predetermined time (such as 24 hours). The event definition list manages itself without any human intervention. If an event is requested again after it has been removed from the event list, the event is reconstructed from the Live Event Database 315.

Votes received by the LEO 355 are cached into the Voting Cache 380 for a predefined time interval and then processed as a batch. In a preferred embodiment, the LEO 355 processes the cached votes every 15 seconds. Alternatively, a different time interval may be selected by a user. The LEO 355 obtains the votes in the batch and tabulates the votes accumulated within the batch for the predefined time interval. These intermediate voting results are then written to the Live Event Database 315. In order to support cross-tabulation of demographic data, raw votes are also written to the Live Event Database 315. The intermediate voting results are cached into memory for all the current events. Periodically, the LEO 355 will refresh the results cached in memory by obtaining the update results from the Live Event Database 355. In a preferred embodiment, the LEO 355 refreshes the voting results approximately every 10 seconds.

FIG. 4 is a general flow diagram of the operation of the high-density interactive voting system shown in FIGS. 2 and 3. In general, the method of the present invention includes facilitating high-density voting over a computer network and tabulating and presenting the final voting results in real-time. More specifically, referring to FIG. 4, the method of the present invention includes creating a survey question (box 400) and receiving a high density

of votes in response to the survey question (box 410). Next, the Live Event Object 355 is provided such that the Object 355 is resident in memory and persistent connections are established and maintained between the Live Event Object 355 and a database (box 420). The database may include, for example, the Live Event Database 315.

The Live Event Object 355 is used to receive and cache the received votes in the Vote Cache 380 (box 430). At a predefined interval, the cached votes are tabulated to generate intermediate voting results (box 440). Final voting results computed by adding all the intermediate voting results (box 450) and these final voting results are display in real time (box 460).

FIG. 5 is a detailed flow diagram illustrating the operation of the Live Event Wizard and the Live Event Staging Component of the Live Event Authoring System shown in FIG. 3. In particular, the Live Event Wizard is used to define an event (box 500) and create polling questions (box 510). The Live Event Staging Component checks the validity of the questions and the event definition (box 520). In addition, the event definition and questions are copied to the database (box 530). At a predetermined time (such as every minute) an inquiry is made as to whether there are new event definitions or questions (box 540). If so, then the database is updated (box 550). Otherwise, the process finishes (box 560) until the next check for new event definition or questions.

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description of the invention, but rather by the claims appended hereto.